Hierarchical and Syntax Structured MT

Ashish Venugopal, Andreas Zollmann

InterACT, LTI, Carnegie Mellon University
Pittsburgh, PA
Stephan Vogel, Alex Waibel

Outline

1. Why pay the Syn - tax
2. Learning Syntax Augmented Grammars
3. Decoding with Syntax Augmented Grammars
4. Widening the S(A)MT pipeline
5. Tools and Conclusion

Why pay the Syn - tax

- Surface form n-gram models are frustrating
  - $P(\text{sweater}|\text{blue}) = \checkmark$
  - $P(\text{sweater}|\text{red}) =?$
  - $P(\text{sweater}|\text{checkered}) =?$
- "Distortion" often distorts sentences
  - Lexical / local distortion
  - Models are too weak to effectively model translation equivalence

Typed Hierarchical Structure

- Model language as a hierarchical, typed process
- Prob. context free grammars rules are natural building blocks
- VP $\rightarrow$ ne $x1$ pas, does not VB_{x1}
  - Example from "What's in a translation rule" Galley et al.

Independence and Constraint

- VP $\rightarrow$ ne $x1$ pas, does not VB_{x1}
- Translation of “ne ... pas” does not depend on words in VB
- Only (and any) VBs can be used in this structure
- Translate + Reorder

Syn CFGs formalism

- Probabilistic Synchronous Context Free Grammars
- $X \rightarrow \langle \gamma, \alpha, \sim, w \rangle$
  - $X \in \mathcal{N}$ is a nonterminal
  - $\gamma \in (\mathcal{N} \cup \mathcal{T}_S)^*$ sequences of $\mathcal{T}_S$, $\mathcal{N}$
  - $\alpha \in (\mathcal{N} \cup \mathcal{T}_T)^*$ sequence of $\mathcal{T}_T$, $\mathcal{N}$
  - $\sim$: $\{1, \ldots, \#\mathrm{NT}(\gamma)\} \rightarrow \{1, \ldots, \#\mathrm{NT}(\alpha)\}$ is a one-to-one nonterminal mapping
  - $w \in [0, \infty)$ is a nonnegative real-valued weight assigned to the rule
- VP $\rightarrow$ does not VB_{x1}, ne $x1$ pas

How do we translate?

- Bottom up chart parsing of source
- Source sequence $\longrightarrow$ nonterminals and associated target translation
- Read translation from resulting parse tree

Decoding

- Initial source sentence

il ne va pas

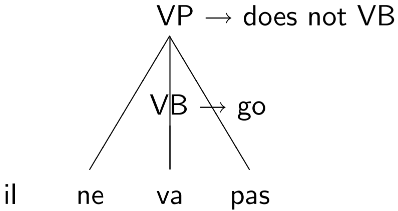
Decoding

VB → go

il ne va pas

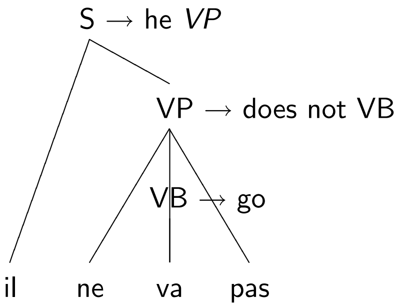
- VB → va, go

Decoding



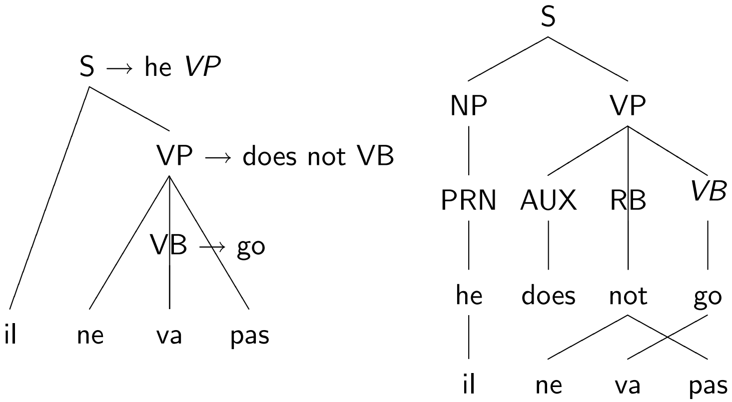
- VP → ne VB_{x1} pas, does not x1

Decoding



- S $\rightarrow$ il VP_{x1} , he x1
- Just one possible derivation!

Categories on Demand: Decoding vs Alignment Graph



What kind of output do you want?

- If you want *real* trees $\cdots$
- Multilevel rules: Tree Substitution Grammars
- Non-contiguous units: Tree Insertion Grammars
  - Example from Chiang, Knight 2006
  - dat Jan Piet de kinderen zag helpen zwemmen
  - that John saw Peter help the children swim
- If you don't care about trees $\cdots$

Flavors of Target Syntax Based MT

- *In the beginning there were* $\cdots$
- Target language parse trees
  - "Syntax-Based" : **tree-driven**
  - Galley 2004, Galley et al. 2006, Marcu et al., 2006
  - Doesn't respect bilingual phrases!
- Phrase pairs, target language parse trees
  - DOP-ish models : **tree-informed**
  - Extract rules from evidence (alignments, parse trees, *phrases*)
  - Chiang 2005, Zollmann 2006
  - Doesn't respect target tree structure

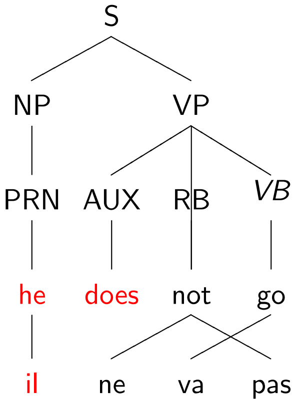
Flavors of Target Syntax Based MT

- *In the beginning there were* $\cdots$
- Target language parse trees
  - "Syntax-Based" : **tree-driven**
  - Galley 2004, Galley et al. 2006, Marcu et al., 2006
  - Doesn't respect bilingual phrases!
- Phrase pairs, target language parse trees
  - DOP-ish models : **tree-informed**
  - Extract rules from evidence (alignments, parse trees, *phrases*)
  - Chiang 2005, Zollmann 2006
  - Doesn't respect target tree structure

Grammar Rule Extraction

- How can we learn probabilistic grammar rules?
- What do we learn them from?
  - French: *Il ne va pas*
  - English: He does not go
  - Phrases (and their spans)
    - *il*, he does
    - *ne va pas*, does not go
- **Goal: Annotate and Compose all initial rules**

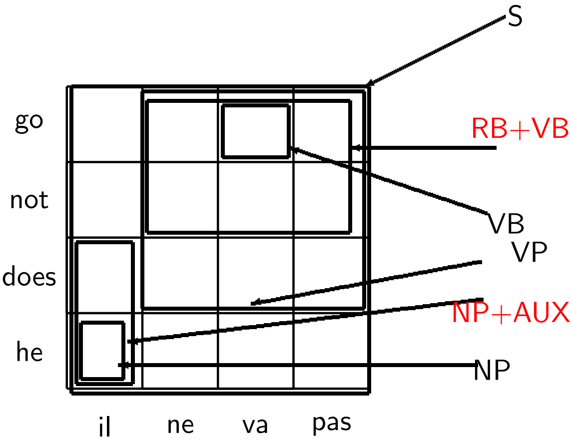
Alignment Graph



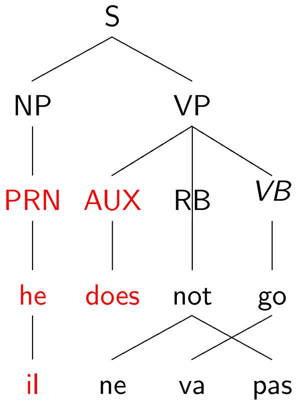
Annotate and Compose

- For each phrase pair, assign a syntactic category based on the target words
- If we can't find a category...
  - CCG style "slash" categories
  - Or 'X+Y' and 'X+Y+Z'
  - Collect evidence from parse tree's base
- Labels can come from anywhere!
- Compose multiple phrase pairs → complex rules.

S $\rightarrow$ he does $RB + VB_{x1}$, il x1



Alignment Graph



- INITIAL+ANNOTATED
  - PRN $\rightarrow$ he, *il*
  - VB $\rightarrow$ go, *va*
  - VP $\rightarrow$ does not go, *ne va pas*
  - S $\rightarrow$ he does not go, *il ne va pas*
- GENERALIZE
  - S $\rightarrow$ he VP_{x1}, il $x1$
  - VP $\rightarrow$ does not VB_{x1}, ne $x1$ pas
  - PRN+AUX $\rightarrow$ il, he does

Sample extracted rules

- S $\rightarrow$ PRN_{x1} ne VB_{x2} pas , x1 does not x2
  - (handles ne pas construction)
- PRN+AUX $\rightarrow$ PRN_{x1} , x1 does
  - (adds an aux in English)
- S $\rightarrow$ $PRN + AUX_{x1}$ $RB + VB_{x2}$, x1 x2
  - (facilitates nonlexical phrase for PRN+AUX)
- RB+VB $\rightarrow$ ne va pas , not go
  - (fully lexicalized construction)
- S $\rightarrow$ $PRN + AUX_{x1}$ ne va pas , x1 not go
  - (facilitates use of PRN+AUX)
- RB+VB $\rightarrow$ ne VB_{x1} pas , not x1
  - (alternative ne pas construction)
- S $\rightarrow$ il ne va pas , he does not go
  - (whole sentence translation)

Decoding with Alternatives

il ne va pas

- Initial source sentence

Decoding with Alternatives

$$\text{VB} \rightarrow \{\text{go,goes,going}\}\ |Cell| = 3$$

il ne va pas

- VB $\rightarrow$ va, go
- VB $\rightarrow$ va, goes
- VB $\rightarrow$ va, going

Decoding

$$\text{VP} \rightarrow \{\text{does not}\},\{\text{no}\}\ \text{VB}\{\text{go,goes,going}\}\ |\mathit{Cell}| = 6$$

$$\text{VB} \rightarrow \{\text{go,goes,going}\}$$

il ne va pas

- $P(\text{go}|\text{does not})$
- $P(\text{go}|\text{not})$
- $\cdots$

Decoding

$\text{S} \rightarrow \{\text{he,it}\}\ VP\{\cdots\}\ |Cell| = 12$

$\text{VP} \rightarrow \{\text{does not}\},\{\text{no}\}\ \text{VB}\{\text{go,goes,going}\}$

$\text{VB} \rightarrow \{\text{go,goes,going}\}$

il ne va pas

- Just one possible derivation (of rules)!

Integration of N-Gram Model

- Integrating N-Gram language model increases the virtual nonterminal space
- Theoretical Runtime: $\mathcal{O}\left(s^3\left[|\mathcal{N}||\mathcal{T}_T|^{2(n-1)}\right]^K\right)$
  - K : maximum number of NT pairs per rule
  - s : source sentence length.
  - $\mathcal{N}$: set of non-terminals
  - $\mathcal{T}$: set of terminals
  - n : order of n-gram LM
- $\mathcal{N} = 38K$ and $n = 3++$

Chart Structure

Each cell i, j contains $\cdots$

- A set of target non-terminal categories X_a, X_b $\cdots$
- Each target non-terminal contains equivalence classes $\cdots$
  - $\langle X_a, t_{left}, t_{right}, i, j \rangle_0$
  - Where each pair t_{left}, t_{right} is unique
- Each equivalence class contains many chart items

Formation of a Chart Item

- Rule: $X_S \rightarrow X_{np}^1\ X_{pp}^2\ X_{vp}^3 \leftrightarrow X_{np}^1\ X_{vp}^3\ X_{pp}^2$
- Example from Zhang et al.
- Terminal Productions: $X_{np}^1\ X_{pp}^2\ X_{vp}^3$
  - $\langle X_{pp}$, [with Sharon], [with Sharon], $i, j\rangle$
  - $\langle X_{pp}$, [in Sharon], [in Sharon], $i, j\rangle$
  - $\vdots$
  - $\langle X_{np}$, [held a], [a meeting], $i, j\rangle$
  - $\langle X_{np}$, [held-up a], [a meeting], $i, j\rangle$
- Number of chart items formed: $\mid X_{np} \mid \times \mid X_{pp} \mid \times \mid X_{vp} \mid$
- We need need to compute LM costs for each permutation

Cube Pruning - Chiang, 2005

- "If an item falls outside the beam, then any item generated using a lower..." $\cdots$
- Only generate the K-Best items of $| X_{np} | \times | X_{pp} | \times | X_{vp} |$
  - Maintains an ordered set of equivalence classes
  - Better K-Best Extraction from Huang, Chiang 2005
  - Optimal K would be retrieved if not for the LM interaction
- *Pruning occurs across rules*
- *Prune away whole equivalence classes!*

Two Pass Decoding

- Two pass decoding:
  - Don't increase virtual nonterminal space during 1st pass
  - Maintain un-explored chart item alternatives during 1st pass
- New Runtime: $\mathcal{O}\left(s^3|\mathcal{N}|^K\right)$
- Search the resulting packed forest for new translations using a left-to-right heuristic search
- Venugopal, Zollmann, Vogel, NAACL 2007
  - Allows integration of flexible, high-order models
  - Limits LM calculations to successful decoding derivations

Decoding

$S \rightarrow \{\text{he},\text{it}\}\ \mathit{VP}\{\text{does not go}\}\ |\mathit{Cell}| = 12$

$\text{VP} \rightarrow \{\text{does not}\},\{\text{no}\}\ \text{VB}\{\text{go},\cancel{\text{goes},\text{going}}\}$

$\text{VB} \rightarrow \{\text{go},\text{goes},\text{going}\}$

il ne va pas

- Only propagate 1 chart item per cell
- Keep the rest of them around for second stage search

Second Stage Search

$\mathrm{S} \rightarrow \{\mathrm{he},\mathrm{it}\}\ \mathit{VP}\{\mathrm{does\ not\ go}\}\ |\mathit{Cell}| = 12$

$\mathrm{VP} \rightarrow \{\mathrm{does\ not}\},\{\mathrm{no}\}\ \mathrm{VB}\{\mathrm{go},\cancel{\mathrm{goes},\mathrm{going}}\}$

$\mathrm{VB} \rightarrow \{\mathrm{go},\mathrm{goes},\mathrm{going}\}$

il ne va pas

- Only propagate 1 chart item per cell
- Keep the rest of them around for second stage search
- Results in a hypergraph of alternative sentence spaning parses

Why Left-to-Right Heuristic Search

- Left-to-right search allows integration of high-order LMs
- This is better than doing N-Best extraction and then re-scoring!
    - See Zollmann, Venugopal 2006 for improvements over re-scoring.

Left-to-Right Heuristic Search for N-Best Items

- Traverse the parse forest in Griebach-Normal Form
- Maintain a sentence spanning beam of trees
- $X_{s0} \longrightarrow X^1_{np}\ X^2_{pp}\ X^3_{vp} \leftrightarrow X^1_{np0}\ X^3_{vp0}\ X^2_{pp0}$
  - $X_{s0} \cdots \leftrightarrow$ Powell $X^3_{vp0}\ X^2_{pp0}$
    - Used X^1_{np1} *update* LM $\mathcal{P}(Powell|\langle s\rangle)$
  - $X_{s0} \cdots \leftrightarrow$ Bowell $X^3_{vp0}\ X^2_{pp0}$
    - Used X^1_{np2} : *update* LM $\mathcal{P}(Bowell|\langle s\rangle)$
  - $\vdots$
  - $|X^1_{np}|$ items added to the beam
  - Factor LM *in* to the real cost
  - Factor *out* the words used in the estimate
  - Update the LM estimate

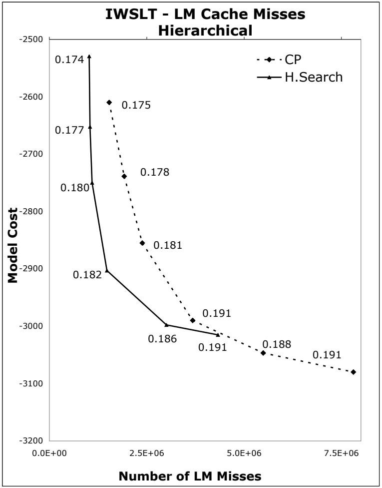
Measuring Impact

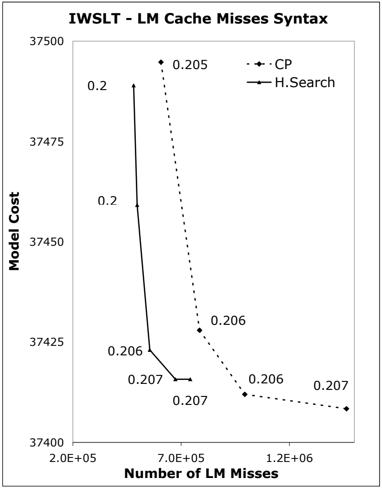
- Two-stage search easily outperforms rescoring/naive pruning
- Cube Pruning vs Two-stage search
  - Evaluate LM cache misses vs Model Cost
  - Evaluate total time vs Model Cost

Experimental Results - Decoding

- IWSLT Evaluation - BTEC travel domain corpus
- 120K Parallel sentences, 1.2M target words
- Eval 500 sentences, average length 10.3 words
- Signficance levels: approx 0.78 BLEU

Two Pass Decoding - LM Cache Misses





SMT pipelines

- SMT systems are component driven
- SAMT: Alignments, Phrase Extraction, Parsing, Rule Extraction
- Each stage is considered as evidence for the next

What does it mean to be evidence?

- Each rule is associated with a feature vector
- Translation = Parsing $\simeq$ Finding best derivation of rules
- $p(D) = \frac{p_{LM}(\mathrm{tgt}(D))^{\lambda_{\mathrm{LM}}} \times \prod_{r \in D} \prod_i \phi_i(r)^{\lambda_i}}{Z(\lambda)}$
- λ learned during MER - not during grammar induction
- ϕ contains MLE and binary/count style features
  - Target word count, IsSyntacticRule, IsBalanced rule etc.

What MLE style features do we use?

- $\hat{p}(r|\,\mathrm{lhs}(X))$: Probability of a rule given its l.h.s category
- $\hat{p}(r|\,\mathrm{src}(r))$: Probability of a rule given its source side
- $\hat{p}(r|\,\mathrm{tgt}(r))$: Probability of a rule given its target side
- $\hat{p}(\mathrm{ul}(\mathrm{src}(r)), \mathrm{ul}(\mathrm{tgt}(r))|\,\mathrm{ul}(\mathrm{src}(r))$: Probability of the unlabeled source and target side of the rule given its unlabeled source side.
- $\hat{p}(\mathrm{ul}(\mathrm{src}(r)), \mathrm{ul}(\mathrm{tgt}(r))|\,\mathrm{ul}(\mathrm{src}(r)))$: Probability of the unlabeled source and target side of the rule given its unlabeled target side.
- Where do the counts come from ?

Softening our notion of evidence

- Extracting a phrase doesn't mean its correct!
- Extracting a rule with such a such a phrase is not correct either?
- What about syntactic categories?
  - Parse "errors" assign incorrect labels?
  - And propagate to incorrect rule arguments!
- We want a distribution over phrase composition,labeling decisions

Reflections on N-Best Lists and Parses

- A phrase from "buggy" alignments is buggy
- A phrase labeled from a "buggy" parse is buggy

- First best parses often contain errors
- Errors are usually the source of variance in n-best lists

Posterior models for MLE feature estimation

- N-Best alignments $a_1, \ldots, a_N$
- GIZA assigned probs $p(a_1 \mid e, f), \ldots, p(a_N \mid e, f)$ renormalized to $\hat{p}(a_i)$
- Same for parses $\hat{p}(\pi_j)$
- $cnt(r) = \sum_{i=1}^{N} \sum_{j=1}^{N'} \hat{p}(a_i) \cdot \hat{p}(\pi_j) \cdot \begin{cases} 1 & \text{if } r \text{ can be extracted from } e,\ f,\ a_i,\ \pi_j \\ 0 & \text{otherwise} \end{cases}$
- Now use $cnt(r)$ in MLE estimates
- Exploit packed structural properties to correctly, efficiently calculate $cnt(r)$

Experimental Results

- IWSLT Evaluation - BTEC travel domain corpus
- GIZA trained to Model 4, Charniak parser 1000 best list
- Initial phrases based on Koehn 2003
- So far, only varied N for alignments vs parses separately

Experimental Results - Lexicon from 1st best, Model 4

N, N'	#Rules	#NTs	Dev	Test	Time
1, 1	300K	1771	23.7	19.8	1145
1, 1	311K	1781	23.7	21.2	1,369
1..5, 1	490K	1894	24.3	21.0	2086
1..10, 1	582K	1947	24.3	20.1	2563
1..25, 1	747K	2026	24.4	20.1	3840
1..50, 1	911K	2072	24.8	21.1	5132
1..10, 1	1m	2212	26.0	22.2	13,406
1, 1..5	616K	2393	23.9	20.0	4291
1, 1..10	850K	2633	24.0	20.1	7237
1, 1..10	652K	2407	25.9	X	13,396

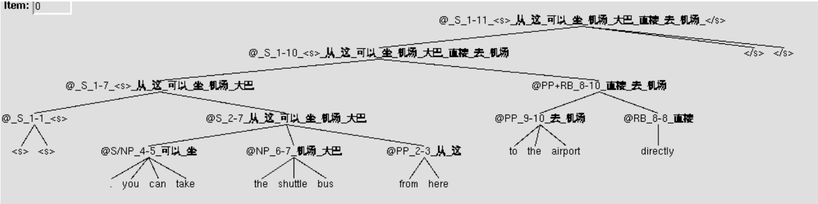
Table: Grammar statistics and translation quality (IBM-BLEU) on development and test set and when integrating N-best alignments an N'-best parses. Decoding time in seconds is on all 500 sentences.

Some interesting rules

- Rules that weren't found in the 1-best list
- IWSLT has non-punctuated source, punctuated targets

count	source	target	LHS NT
247.93	请	please .	@UH+.
210.69	请	please .	@VB+.
162.06	想	'd	@MD
153.42	我	, I	@,+PRP
146.32	我	I have	@PRP+AUX
141.96	我	.	@.
141.75	的	in	@IN

System Output



System track record

- Beating or matching phrase based baselines
- Small and medium data tasks
- Chinese-English IWSLT
- (French/Spanish)-English Europarl
- Chinese-English NIST

IWSLT Chinese English

Rules	Dev IBM-BLEU	Test IBM-BLEU
X grammar	21.25	18.08
Pharaoh	22.0	19.3
SAMT	23.50	20.04

Table: *Comparison of translation-models system using "SmartCase", evaluated on the official case and punctuation sensitive IBM-BLEU metric*

Spanish-English

- 2000 sentences Test 06 Spanish English Europarl
- PhraseBased: 31.76
- SyntaxAugmented: 32.15
- Minimal impact of Re-ordering for Spanish
  - Development data (tuned)
  - Window 1: 31.98
  - Window 2: 32.24
  - Window 3: 32.30
  - Window 4: 32.26
  - Syntax: 32.48

Chinese-English NIST

- Chinese-English NIST Evaluation - 1 day worth of training time - 3-gram LM on target side of data
- Case Sensitive Offical NISTBLEU
- No. Rules applicable to Dev and Test.
  - X: Style of Chiang 2005
  - Penn: Retains only those that are constituents
  - CCG+: Assigns categories to almost all lexical phrases

Grammar	NTs	Rules	Time	Dev (MT03)	Test (MT05)
X	2	197K	1.9h	23.5	X
Penn	73	191K	0.3h	22.8	21.1
CCG+	38,861	795K	0.9h	28.7	26.2

Open Source Tools

- All tools available at www.cs.cmu.edu/žollmann/samt/
- *extractrules.pl* - identify Syn CFG rules
- *fiilterrules.pl* - score and prune rules
- *FastTranslateChart* - Chart parser decoder, N-best lists, MER
- *MER* - standalone MER toolkit